## REMARKS

Applicants respectfully request favorable reconsideration of this application, as amended. Applicants would also like to thank Examiners Colin and Moise for the courtesies extended to Applicant's representative, Mr. Jason Vick, during the July 21 Personal Interview. During the interview, Claims 15-17 were discussed as well as a summary of the invention provided with reference to the figure in the application. More particularly, it was pointed out that an isolation means is between the input/output module and the encryption module as illustrated in the figure.

The Office Action objects to the specification and a number of claims for various informalities. Attached hereto is a substitute specification with the lines double spaced on good quality paper as requested by the Examiner. Additionally, the claims have generally been amended to provide clear antecedent basis and to overcome the objections raised by the Examiner.

As pointed out during the Personal Interview, an isolation means is provided between the input/output module and the encryption module as recited in Claim 15. The isolation means make sensitive information stored in the encryption module inaccessible to the host computer system and ensures parallelism of the operations performed by the input/output module and the encryption module.

Bakhle, which was used in support of the rejection under 35 U.S.C. § 102(e), fails to teach or suggest this feature. Moreover, the IBM Technical Disclosure Bulletin, which was used in support of the rejection under 35 U.S.C. § 103 fails to overcome the deficiencies as noted above in relation to Bakhle.

Accordingly, since the references of record, either alone or in combination, fail to teach or suggest each and every feature of the claims, the references fail to render obvious or anticipate the claimed subject matter.

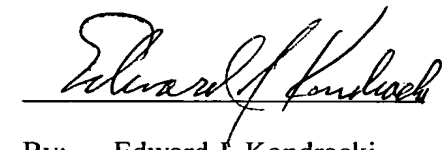
With all objections and rejections being overcome, an early and favorable Notice of Allowance is respectfully requested.

Should the Examiner believe anything further is desirable in order to place the application in even better condition for allowance, the Examiner is encouraged to contact Applicant's undersigned representative at the telephone number listed below.

The Commissioner is hereby authorized to charge to deposit account number 50-1165 (Docket No. T2147-906625) and fees not included herein, under 37 CFR §§ 1.16 and 1.17, that may be required by this paper and to credit any overpayment to that Account. A duplicate copy of this page is included for such purpose. If any additional extension of time is required in connection with the filing of this paper and has not been separately requested, such extension is hereby requested.

Respectfully submitted,

MILES & STOCKBRIDGE P.C.



Date: July 27, 2004

By: Edward J. Kondracki
Registration No. 20,604

Miles & Stockbridge P.C.
1751 Pinnacle Dr., Suite 500
McLean, VA 22102
Phone 703-610-8627
Fax 703-610-8686

# ARCHITECTURE OF AN ENCRYPTION CIRCUIT IMPLEMENTING VARIOUS TYPES OF ENCRYPTION ALGORITHMS SIMULTANEOUSLY WITHOUT A LOSS OF PERFORMANCE

The present invention applies to the field of encryption, and more particularly, relates to an architecture of an encryption circuit implementing various types of encryption algorithms simultaneously.

This architecture is embodied by a circuit supported by a PCI (Peripheral Component Interconnect) card, and makes it possible to implement various encryption algorithms in parallel, without a loss of performance in a machine (server or station). It also plays the role of a vault in which the secret elements (keys and certificates) required for any electronic encryption function are stored.

The increased need for performance in cryptography, combined with the need for inviolability has led the manufacturers of security systems to favor hardware solutions in the form of additional cards.

Such a card, coupled with a server, constitutes the hardware security element of the server.

There are known implementations of security architectures based on ASIC (Application Specific Integrated Circuit) components, which entail high development costs for a solution that remains inflexible, both on the manufacturer end and on the user end.

Furthermore, there is no architecture existing today that is capable of executing a set of algorithms simultaneously with a guaranteed throughput for each of them.

The object of the invention is specifically to eliminate the aforementioned drawbacks and to meet the market's new demands for security.

To this end, the subject of the invention is an architecture of an encryption circuit simultaneously processing various encryption algorithms, the circuit being capable of being coupled with a host system hosted by a computing machine.

According to the invention, the circuit comprises:

- an input/output module responsible for the data exchanges between the host system and the circuit via a PCI bus;

- an encryption module coupled with the input/output module, in charge of the encryption and decryption operations as well as the storage of all of the circuit's sensitive information; and

- isolation means between the input/output module and the encryption module, making the sensitive information stored in the encryption module inaccessible to the host system, and ensuring the parallelism of the operations performed by the input/output module and the encryption module.

The first advantage of the invention is that it allows fast execution of the principal encryption algorithms with two levels of parallelism, a first parallelism of the operations performed by the input/output module and the encryption module, and a second parallelism in the execution of the various encryption algorithms.

Another advantage of the invention is to make invisible to the host system all of the encryption resources made available to the system, and to provide protected storage for secrets such as keys and certificates. The sensitive functions of the card (algorithms and keys) are all located inside the encryption module and are inaccessible from the PCI bus.

The invention also has the advantage of enabling hardware and software implementations of various encryption algorithms to coexist without a loss of performance, while guaranteeing the throughputs of each of them.

It has the further advantage of being scalable by a choice of standard microprocessor and programmable logic technologies, as opposed to more conventional implementations based on specific circuits (ASIC). The invention makes it possible, in particular, to implement proprietary algorithms simply by modifying the code of the encryption processors or by loading a new configuration file for the encryption automata of the encryption module.

Other advantages and characteristics of the present invention will emerge through the reading of the following description, given in reference to the attached figure, which represents a block diagram of an architecture according to the invention.

For simplicity's sake, the encryption/decryption module will hereinafter be called the "encryption module."

The links between each module are all two-way links unless indicated.

The encryption circuit 1 according to the invention hinges on two main modules:

- an input/output module 2 responsible for the data exchanges between the encryption resources and a host system HS via a PCI bus; and

- an encryption module 3 in charge of the encryption and decryption operations as well as the storage of the secrets.

These two modules 2 and 3, respectively delimited by an enclosing dot-and-dash line, dialogue via a dual-port memory DPR 4 that allows the exchange of data and commands/statuses between the two modules 2 and 3.

A serial link SL controlled by the encryption module 3 also makes it possible to input the basic keys through a secure path SP independent of the normal functional path (PCI bus), thus meeting the requirement imposed by the FIPS140 standard.

This link SL is connected to the card 1 via a module EPLD 5, or "Erasable Programmable Logic Device," coupled between the input/output module 2 and the encryption module 3, that ensures logical consistency between the modules.

The input/output module 2 includes the following elements:

- a microcontroller IOP 6 primarily constituted by a processor $6_1$ and by a PCI interface $6_2$, integrating DMA (Direct Memory Access) channels. These are channels that are specific, or dedicated, to the processor, through which the data exchanged between the memories passes, and which are coupled with the processor without using the resources of the processor;

- a flash memory 7, which is a memory that saves the stored data without a power source and whose storage capacity is for example 512 kilobytes; and

- an SRAM memory 8, from the abbreviation for "Static Random Access Memory" which is a memory that requires a power source in order to save the data stored in the memory, and whose storage capacity is for example 2 Megabytes.

The data transfers between the encryption module 3 and the host system HS take place simultaneously with the encryption operations performed by the encryption module 3, thus making it possible to optimize the overall performance of the card 1.

The flash memory 7 contains the code of the processor of the microcontroller IOP 6.

At startup, the processor copies the contents of the flash memory 7 into the SRAM memory 8; the code being executed in this memory for better performance.

The SRAM memory 8 could also be replaced by an SDRAM (Synchronous Dynamic RAM) memory, which is a fast dynamic memory.

The microcontroller IOP 6 is capable of managing this type of memory without a loss of performance.

The choice of the microcontroller depends primarily on the desired performance objectives as well as the total power consumption of the card supporting the circuit, which is generally limited to 25 W (PCI specification).

The dual-port memory DPR 4 provides the isolation between the input/output module 2 and the encryption module 3, thus making the latter inaccessible to the host system HS.

Its storage capacity in the example described is 64 kilobytes. It temporarily stores the data that is to be encrypted or decrypted by the encryption automata of the encryption module 3.

It is divided into two areas:

- a control area, for example of 4 kilobytes, in which the microcontroller IOP 6 writes the control blocks to be sent to the automata; and

- a data area, for example of 60 kilobytes, containing the data to be processed by the automata.

The encryption module 3 includes first and second encryption sub-modules $3_1$ and $3_2$, respectively delimited by an enclosing broken line.

The first sub-module $3_1$ includes an SCE (Symmetric Cipher Engine) component 9, dedicated to the processing of symmetric encryption algorithms, coupled with the bus of the dual-port memory 4.

The second sub-module $3_2$ is dedicated to the processing of asymmetric encryption algorithms.

It is coupled with the bus of the dual-port memory 4, and includes a separate internal bus isolated from the bus of the dual-port memory 4.

It also includes:

- one or two processors CIP $10_1$, $10_2$, from the abbreviation for "Cipher Processor";

- a processor ACE $10_2$, from the abbreviation for "Asymmetric Cipher Processor," which in a variant of embodiment replaces one of the two cipher processors CIP $10_1$, $10_2$;

- a CMOS memory 11, for example with a storage capacity of 256 kilobytes, backed up by a battery;

- a flash memory PROM 12, from the abbreviation for "Programmable Read-Only Memory," for example with a storage capacity of 512 kilobytes; and

- an SRAM memory 13, for example with a storage capacity of 256 kilobytes.

As illustrated in the block diagram of the figure, the SCE component 9 and the CMOS memory 11 are directly coupled with the bus of the dual-port memory DPR 4, while the processors CIP $10_1$ and $10_2$ and the flash 12 and SRAM 13 memories are coupled with a separate bus isolated from the bus of the dual-port memory DPR 4 by means of a bus isolator 14, also called a bus "transceiver," represented in the figure by a block with two opposing arrows.

The flash memory PROM 12 located in the bus of the processors CIP $10_1$ and $10_2$ contains all of the software used by the encryption module 3.

The SRAM memory 13 plays two roles:

- it enables the fast execution of the code of the processors CIP $10_1$ and $10_2$; the code is copied into the memory from the flash memory PROM 12 at power up;

- it also makes it possible to store the data temporarily during the execution of the algorithms.

This characteristic of the architecture guarantees the independence of the various encryption automata from one another.

The processor CIP $10_1$ and the processor ACE $10_2$ both access the dual-port memory DPR 4 in order to read or write the data to be encrypted, but the processing of the algorithms *per se* takes place entirely within their own memory space (internal cache and SRAM 13) without interfering with the SCE component 9.

The SCE component 9 integrates the various symmetric encryption automata (one automaton per algorithm) of the DES, RC4 or other type, as well as a random number generator, not represented.

Each automaton works independently from the others and accesses the dual-port memory DPR 4 in order to read its control block (written by the microcontroller IOP 6) and the corresponding data to be processed.

The parallelism of the processing thus performed makes it possible to guarantee an optimal throughput for each algorithm even when the automata are used simultaneously.

The only limitation on the processing is imposed by access to the dual-port memory DPR 4, which is shared by all of the automata.

The bandwidth of the data bus to this memory must therefore be greater than the sum of the throughputs of each algorithm in order not to limit their performance.

The SCE component 9 is produced using a programmable technology that is also known as FPGA, or "Field Programmable Gate Array," which is a programmable circuit or chip having a high logic gate density, which provides all of the flexibility required to implement new algorithms, including proprietary algorithms, on demand.

The configuration data for this component is contained in the flash memory PROM 12, and is loaded into the SCE component 9 at power up under the control of the processor CIP $10_1$.

The processor CIP $10_1$, using given programming software, implements the algorithms not implemented in the SCE component 9. It also implements asymmetric algorithms of the RSA type with or without the help of the specialized automaton implemented by the processor ACE $10_2$.

It performs the initialization of the security parameters (keys) via the serial link SL.

The utilization of a high-performance processor at this level guarantees optimal performance in the execution of the algorithms as well as great flexibility for the implementation of additional algorithms.

As a result of this processor, it is also possible to download proprietary algorithms via the serial link SL.

According to a first embodiment, two processors CIP $10_1$ and $10_2$ are implemented:

One of them $10_1$ is required for the execution of the of the RSA algorithm; the other $10_2$ implements the algorithms not yet supported by the SCE component 9.

According to a second embodiment, there is only one processor CIP $10_1$ assisted by a processor ACE $10_2$ that replaces one of the two processors CIP $10_1$ and $10_2$ of the first embodiment, and which implements, in programmable logic, the intensive calculation linked to the protocol of the RSA algorithm.

All of the required algorithms are implemented in programmable logic in automata of the SCE component 9.

This component is produced in programmable FPGA technology.

The CMOS memory 11 contains the keys and other secrets of the card 1. It is backed up by a battery and protected by various known security mechanisms SM 15 which, in case of abnormalities, translate them as an intrusion attempt and erase its contents.

These abnormalities are for example due to:

- an abnormal increase or decrease in the temperature;
- an abnormal increase or decrease in the supply voltage;
- a disencryption of the card;
- a physical intrusion attempt (on the card end or the host system end);
- etc.

Each of the above events triggers an alarm signal that acts on the reset mechanism of the CMOS memory 11.